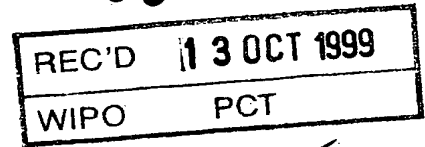


09/744522

**PRIORITY
DOCUMENT**

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)



DE 99 / 2086

Bescheinigung

Die Siemens Aktiengesellschaft in München/Deutschland hat eine Patentanmeldung
unter der Bezeichnung

"Schaltungsanordnung und Verfahren zur Daten-
konvertierung in einem Prozessor"

am 31. Juli 1998 beim Deutschen Patent- und Markenamt eingereicht.

Die angehefteten Stücke sind eine richtige und genaue Wiedergabe der ursprüng-
lichen Unterlagen dieser Patentanmeldung.

Die Anmeldung hat im Deutschen Patent- und Markenamt vorläufig das Symbol
G 06 F der Internationalen Patentklassifikation erhalten.

München, den 25. August 1999

Deutsches Patent- und Markenamt

Der Präsident

Im Auftrag

Keller

Zeichen: 198 34 632.8

This Page Blank (uspto)



Beschreibung

Schaltungsanordnung und Verfahren zur Datenkonvertierung in einem Prozessor

5

In Prozessoren oder Mikroprozessoren wird vor arithmetischen oder logischen Operationen mit unterschiedlichen Datentypen durch Programmsequenzen eine Datentypkonvertierung herbeigeführt. Die Datentypkonvertierung mittels Programmsequenzen

10

hat jedoch den Nachteil, daß sie die Verarbeitungsgeschwindigkeit eines Prozessors oder Mikroprozessors herabsetzt. Desweiteren weist diese Art der Datentypkonvertierung den Nachteil auf, daß das Bussystem des Prozessors durch die zur Datentypkonvertierung erforderlichen Operations-Code zusätzlich belastet wird.

15

Neben der angesprochenen Datentypkonvertierung werden objekt-orientierte Programm Sprachen eingesetzt, um spezielle Probleme zu lösen. Eine Datentypkonvertierung kann auch durch eine objekt-orientierte Befehlsstruktur erreicht werden. Die objekt-orientierte Befehlsstruktur bringt jedoch den Nachteil einer Vergrößerung des Programmcodes mit sich, da jeder Befehl für jede Kombination von Datentypen im Speicher hinterlegt sein muß. Eine Vergrößerung des Programmcodes hat ebenfalls eine Verringerung der Verarbeitungsgeschwindigkeit zur Folge.

20

25

Der Erfindung liegt die Aufgabe zugrunde, eine Schaltungsanordnung und ein Verfahren zur Datentypkonvertierung anzugeben, die die oben genannten Nachteile vermeidet.

30

Die Aufgabe wird durch die Merkmale des Patentanspruchs 1 und 8 gelöst.

35

Die Erfindung bringt den Vorteil mit sich, daß eine Datenkonvertierung ohne eine Verringerung der Verarbeitungsgeschwindigkeit des Prozessors durchgeführt werden kann.

Die Erfindung bringt den Vorteil mit sich, daß Datentypumwandlungen selbständig durchgeführt werden.

- 5 Die Erfindung bringt den weiteren Vorteil mit sich, daß der Programmcode nicht vergrößert werden muß.

10 Die Erfindung bringt den weiteren Vorteil mit sich, daß eine datentypgerechte Datentypumwandlung mit einer datentypgerechten Adreßberechnung durchgeführt wird.

Weitere Besonderheiten sind in den Unteransprüchen angegeben.

15 Die Schaltungsanordnung und das Verfahren werden aus der nachfolgenden näheren Erläuterung zu einem Ausführungsbeispiel anhand von Zeichnungen ersichtlich.

Es zeigen:

- 20 Figur 1 den Aufbau einer Objektadresse,
Figur 2 den Aufbau eines Registers,
Figur 3 den Aufbau interner Register,
Figur 4 eine Ausgestaltung eines Prozessors und
25 Figur 5 die Ausgestaltung des Prozessors mit einer objekt-orientierten Datentypkonvertierung entsprechend der Erfindung.

In Figur 1 ist eine Aufteilung einer Objektadresse OA gezeigt. Diese Objektadresse OA unterteilt sich in einen Bereich für die Angabe des Typs T des Objekts und eine zum Objekt gehörende Speicheradresse SA. Die Speicheradresse SA gibt den Speicherplatz eines Objektes in einem Speicherbereich eines Speichers an. Der Objekttyp T und die Speicheradresse SA werden bei einer Deklaration des Objektes von
30 einem Compiler vergeben und als eine Einheit behandelt. Objekte können beispielsweise Daten, Adressen oder Codeadressen sein.
35

In Figur 2 ist die Aufteilung eines Registers R wiedergegeben. Dieses Register R ist in einen ersten Bereich zur Angabe des Daten- bzw. Objekttyps T und in einen zweiten Bereich zur
5 Abspeicherung der Daten bzw. Objekte D aufgeteilt.

In Figur 3 ist eine detaillierte Angabe der Aufteilung des Registers R und der Objektadresse OA wiedergegeben. In dieser Darstellung werden die abzuspeichernden Adressen oder Daten
10 wie oben angedeutet in einen ersten und zweiten Bereich unterteilt. In einem Register R mit einer Länge von 35 Bit, sind beispielsweise 3 Bit für die Angabe des Objekttyps T und 32 Bit für das abzuspeichernde Objekt, die Adresse oder Daten vorgesehen. Die 32 Bit für das Objekt können ein Datenwort,
15 eine Daten- oder Code-Adresse oder eine Speicheradresse mit einer Angabe über den Typ der Speicheradresse sein. Weitere Unterteilungen sind in Figur 3 dargestellt.

In der Figur 4 ist ein Prozessor, insbesondere ein RISC-Prozessor dargestellt. In dieser Darstellung sind die wesentlichen, in einer Pipeline angeordneten Komponenten des RISC-Prozessors ein Sequenz-Controller SC, ein Instruction-Decoder ID, eine Register-Read-Einheit RR, eine Execute-Einheit E, eine Data-Transfer-Einheit DT und eine Register-Write-Einheit
25 RW. Daten werden entweder von einer externen Speichereinheit M über die Data-Transfer-Einheit DT ein- oder ausgelesen. Daten, die für den laufenden Verarbeitungsprozeß von Bedeutung sind, werden über die Register-Write-Einheit RW in einem Register-File RF abgelegt und nach Bedarf wieder durch die Register-Read-Einheit RR gelesen. In der Execute Einheit E werden
30 logische und arithmetische Befehle ausgeführt.

In Figur 5 ist der RISC-Prozessor mit mindestens einer Datentypkonvertierung ausführenden Datentyp-Konvertierungseinheit TC ausgebildet. Diese Datentyp-Konvertierungseinheit
35 kann beispielsweise vor und nach der Execute-Einheit E oder zwischen der externen Speichereinheit M und der Data-Trans-

fereinheit DT angeordnet sein. Die Datentyp-Konvertierungseinheit kann ebenso zwischen der Register-Write-Einheit RW und dem Register-File RF und zwischen dem Register-File RF und der Register-Read-Einheit RR angeordnet werden, um eine
5 notwendig werdende Datenkonvertierung durchzuführen. Die gezeigte Anordnung mit einer Datentyp-Konvertierungseinheit TC in der Datenpipeline des RISC-Prozessors bringt den Vorteil mit sich, daß die Taktfrequenz der Pipeline unverändert bleibt.

10

In der vorliegenden Erfindung wird die Objektadresse OA in einen ersten und zweiten Bereich unterteilt. Aus dem ersten Bereich ermittelt die Datentyp-Konvertierungseinheit TC den Datentyp T und aus dem zweiten Bereich die physikalische
15 Adresse SA.

20

Bei einem Speicherzugriff wird eine Adressenberechnung von indizierten Load- und Store-Befehlen mit Hilfe des Objekttyps ein Adreßoffset entsprechend der Objektgröße für die physikalische Adresse SA ermittelt.

25

Bei einem Load-Befehl ist die Speicheradresse des Objektes bekannt. Das Objekt wird mit der Angabe des Typs in einem Register des Prozessors abgelegt.

30

Bei einem Store-Befehl werden die zu speichernden Daten und der dazugehörige Objekttyp aus den Register entnommen, in den Objekttyp der Zieladresse umgewandelt und unter der Zieladresse abgelegt.

35

Execute-Befehle enthalten neben den Registernummern von Quell- und Ziel-Registern auch den zugehörigen Objekttyp für das Zielregister. Der Objekttyp der Operation wird vom Objekttyp des Zielregisters abgeleitet. Eine Verarbeitungseinheit in der Execute-Einheit E für beispielsweise eine arithmetische Operation wird anhand des Objekttyps ausgewählt. Da vor einer arithmetischen Operation die Objekttypen der Quell-

und Ziel-Register bekannt sind, kann eine Datentypumwandlung entsprechend vorgenommen werden. Vor einer Operation werden die Quell-Daten in einen Datentyp umgewandelt, der von einer Execute-Einheit E eines RISC-Prozessors verarbeitbar ist.

- 5 Nach einer Operation kann das Ergebnis in den Datentyp eines Zielregisters umgewandelt werden.

Bei einer Multiplikation von einer ersten und zweiten Variablen werden durch den Compiler jeweils die physikalische Adresse und der Datentyp generiert. Beim Laden der ersten und zweiten Variablen werden die Datentypen im Register-File RF abgelegt. Der Programmabschnitt - Load Variable 1 - signalisiert dem Prozessor die Größe der ersten Variablen, beispielsweise ein 8 Bitwert oder 16 Bitwert. Der Datentyp-Konvertierungseinheit TC wird mit dem Datentyp mitgeteilt, ob es sich um eine signed-Integer-Variable oder eine unsigned-Integer-Variable handelt. Nach dem Laden der ersten Variablen werden die zur zweiten Variablen gehörenden Daten geladen. Nach dem beim Lesen der Daten aus dem Register-File RF die Datentyp-Konvertierungseinheit TC die beiden Datentypen der ersten und zweiten Variablen kennt, werden die Datentypen einander angepaßt und die eigentliche Operation, wie beispielsweise eine Multiplikation in der nachgeordneten Execute-Einheit E ausgeführt.

Die Datentyp-Konvertierungseinheit TC kann nach der Execute-Einheit E in die Datenpipeline des RISC-Prozessors an verschiedenen Stellen eingeführt werden. Ein Fehlen der Datentyp-Konvertierungseinheit TC nach der Execute-Einheit E hat zur Folge, daß die Ergebnisdaten, die in der Execute-Einheit E gebildet wurden, zusammen mit einem sich aus der jeweiligen Operation ergebenden Datentyp im Register-File RF abgelegt werden.

Bei einem Store-Befehl werden sowohl die Daten als auch der jeweilige Datentyp wieder aus dem Register-File RF ausgelesen und vor der Execute-Einheit E einer Datentyp-Konvertierungs-

einheit TC zugeführt. Dort kann die Datentyp-Umwandlung entsprechend dem Datentyp der Zieladresse erfolgen.

Bei einer indizierten Adressierung braucht der Datentyp-Konvertierungseinheit TC nur der einfache Index mitgeteilt zu werden, und der Prozessor kann sich anhand des Datentyps der Adresse den Byte-Offset errechnen. So ist beispielsweise bei einem normalen Integer-Wert, der eine Länge von 2 Byte aufweist, bekannt, daß der Index entsprechend verdoppelt werden muß und zur physikalischen Adresse dazu addiert werden muß. Dies bringt den Vorteil mit sich, daß zeitaufwendige Umrechnungen zur Ermittlung der Adreß-Indizes entfallen.

Ein weiterer Vorteil der indizierten Adreßberechnung liegt darin, daß mit ein und demselben Programmcode eine Tabelle von Daten mit short Integer-Werten oder long Integer-Werten oder normalen Integer-Werten bearbeitet werden kann. Der Programmcode braucht dabei die genannten Datentypen nicht zu unterscheiden, da der Prozessor sich seine Adressen anhand der Datentypen selbst berechnet.

Patentansprüche

1. Schaltungsanordnung zur Datenkonvertierung in einem Prozessor mit mindestens einer logischen oder arithmetischen Operation ausführenden Einheit (E),
dadurch gekennzeichnet,
daß eine objektorientierte Datenkonvertierungseinheit (TC) zur Erkennung eines Typs (T) eines Objektes (D) und einer Objektadresse (OA) vor der logischen oder arithmetischen Operation ausführenden Einheit (E) angeordnet ist, und aufgrund der mit einer Objektadresse (OA) und dem Objekt (D) mitgeführten Typinformation die Datenkonvertierungseinheit (TC) den Typ (T) des Objektes (D) erkennt und vor einer Operation die Objekte (D) bei einer Ungleichheit angleicht oder einen vorbestimmten Typ des Objektes (D) erzeugt.
2. Schaltungsanordnung nach Anspruch 1,
dadurch gekennzeichnet,
daß ein Speicherplatz für eine Objektadresse (OA) und ein Speicherplatz eines Registers (R) jeweils in einen ersten und zweiten Bereich (T, SA; T, D) aufgeteilt ist, wobei im ersten Bereich jeweils ein Typ (T) des Objektes (SA, D) abgelegt ist.
3. Schaltungsanordnung nach Anspruch 1,
dadurch gekennzeichnet,
daß die objektorientierte Datenkonvertierungseinheit (TC) nach der logischen oder arithmetischen Operation ausführenden Einheit (E) vorgesehen ist.
4. Schaltungsanordnung nach Anspruch 1,
dadurch gekennzeichnet,
daß die objektorientierte Datenkonvertierungseinheit (TC) vor dem Abspeichern des Objektes (D) in einem externen Speicher (M) und einem Register-File (RC) angeordnet ist.

5. Schaltungsanordnung nach Anspruch 1,
dadurch gekennzeichnet,
daß ein Register-File (RC) in einen Speicherbereich für Daten
und einen Speicherbereich für eine jeweilige Typangabe der
5 Daten unterteilt ist.

6. Schaltungsanordnung nach Anspruch 1,
dadurch gekennzeichnet,
daß dieser ein Reduced Instruction Set Computer (RISC) ist.

10

7. Schaltungsanordnung nach Anspruch 1,
dadurch gekennzeichnet,
daß dieser ein Complex Instruction Set Computer (CISC) ist.

15 8. Verfahren zur Datenkonvertierung in einem Prozessor mit
mindestens einer eine logische oder arithmetische Operation
ausführenden Einheit (E),
dadurch gekennzeichnet,
daß durch eine Typinformation (T) in einer Objektadresse (OA)
20 und durch eine Typinformation (T) eines Objektes (D) eine
objektorientierte Datenkonvertierung durchgeführt wird und
bei einer Ungleichheit der durch eine logische oder arithme-
tische Operation zu verknüpfenden Objekte (D) der Typ der Ob-
jekte einander angeglichen oder ein vorbestimmter Objekttyp
25 eines Objektes (D) erzeugt wird.

9. Verfahren nach Anspruch 8,
dadurch gekennzeichnet,
daß ein Speicherplatz für eine Objektadresse (OA) und ein
30 Speicherplatz eines Registers (R) jeweils in einen ersten und
zweiten Bereich (T, SA; T, D) aufgeteilt wird und im ersten
Bereich (T) jeweils eine Typinformation der im zweiten Be-
reich der Objektadresse (OA) abgelegten Speicheradresse (SA)
und der im zweiten Bereich abgelegten Daten (D) des Registers
35 (R) vermerkt wird.

Zusammenfassung

Schaltungsanordnung und Verfahren zur Datenkonvertierung in einem Prozessor

5

Eine in einem Prozessor durchzuführende Datenkonvertierung wird objektorientiert vor einer arithmetischen oder logischen Operation jeweils auf der Grundlage einer zu dem Objekt gehörenden Datentypangabe durchgeführt.

10

Figur 3

This Page Blank (uspto)

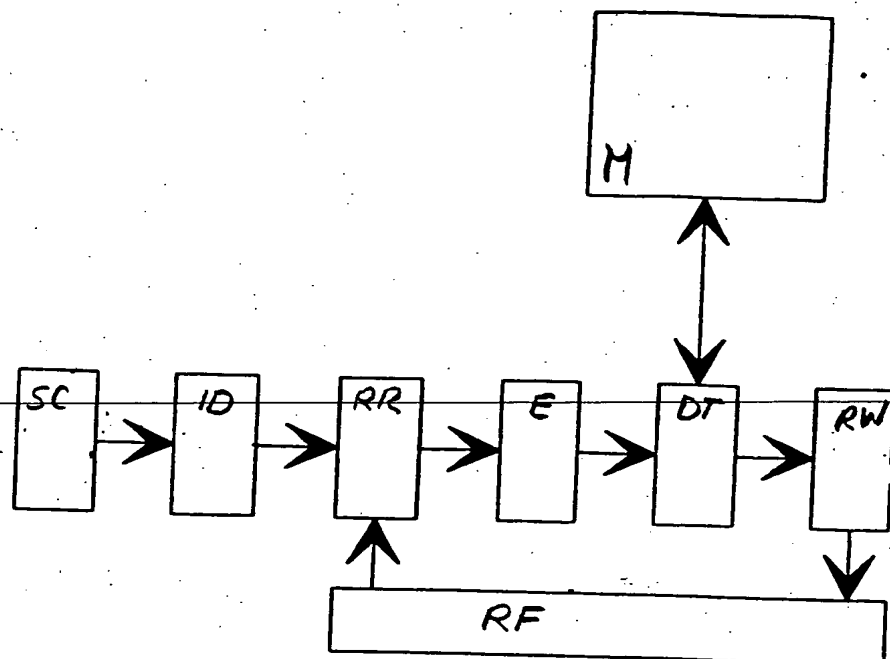


Fig. 4

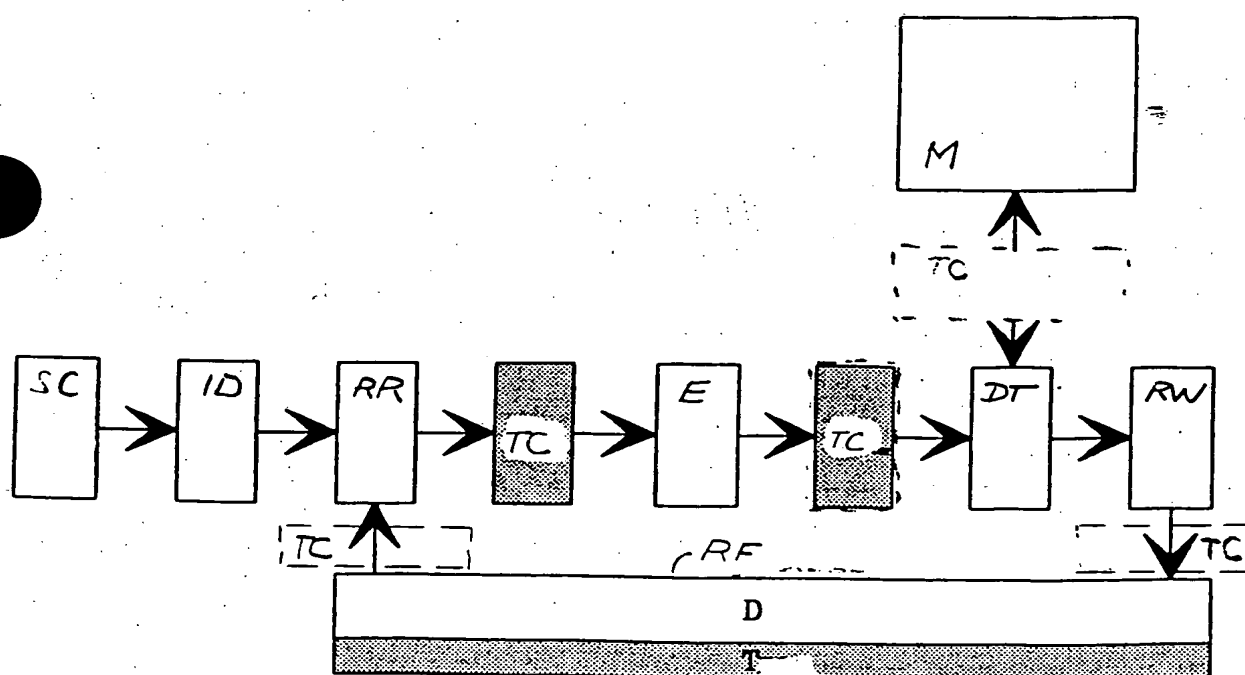


Fig. 5

This Page Blank (uspto)

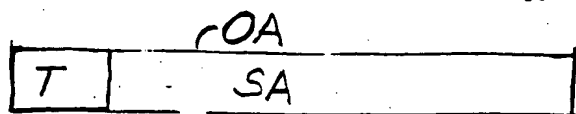


Fig 1

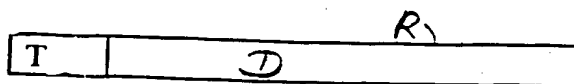


Fig 2

36 Bit Register:

Address:

Object Address:

unsigned Character:

signed Character:

unsigned Integer:

signed Integer:

signed long Integer:

signed long Integer:

Type 3 Bit		Register Data 32 Bit	
0		Data/Code Address 32 Bit	
1		Type 3Bit	Memory Address 29 Bit
2		Data	
3		T	Data
4		Data	
5		T	Data
6		Data	
7	T	Data	

Fig 3

This Page Blank (uspto)